	Application No.	**Applicant(s)**
Examiner-Initiated Interview Summary	10/660,166	DAY ET AL.
	Examiner	**Art Unit**	
	Michael D. Pham	2167	

All Participants:

(1) *Michael D. Pham*.

(2) _____.

Status of Application: *Pending*

(3) *Scott A. Stinebruner*.

(4) _____.

Date of Interview: *30 August 2007*

Time: *11am*

Type of Interview:
- ☒ Telephonic
- ☐ Video Conference
- ☐ Personal (Copy given to: ☐ Applicant   ☐ Applicant's representative)

Exhibit Shown or Demonstrated: ☐ Yes   ☐ No
If Yes, provide a brief description:

Part I.

Rejection(s) discussed:

Claims discussed:
Independent claims

Prior art documents discussed:

Part II.

SUBSTANCE OF INTERVIEW DESCRIBING THE GENERAL NATURE OF WHAT WAS DISCUSSED:
Dis■cussed Amendments to clarify claims in order to put the case into condition for allowance.

Part III.

☒ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview directly resulted in the allowance of the application. The examiner will provide a written summary of the substance of the interview in the Notice of Allowability.

☐ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview did not result in resolution of all issues. A brief summary by the examiner appears in Part II above.

(Examiner/SPE Signature)

(Applicant/Applicant's Representative Signature – if appropriate)

Fax Cover Sheet

Date: 29 Aug 2007

To: Scott A. Stinebruner	**From:** Michael D. Pham
Application/Control Number: 10/660,166	**Art Unit:** 2167
Fax No.: 513-241-6234	**Phone No.:** (571)272-3924
Voice No.: 507-253-5333	**Return Fax No.:** (571) 273-8300
Re:	**CC:**

☐ **Urgent**   ☐ **For Review**   ☐ **For Comment**   ☐ **For Reply**   ☐ **Per Your Request**

Comments:
Attached is a proposed amendment.   The limitations have been moved around for further clarity.

Number of pages 3 **including this page**

Claims 5-7 canceled.
Claims 1, 8 amended.
Claim 24 added.

1. (Currently Amended) A method for monitoring a query during runtime, said query involving a plurality of join operations, the method comprising the steps of:

<u>wherein the plurality of join operations include a first join that includes a first table and a second table and a second join that includes the first table and a third table;</u>

running the query according to a first join order;

generating a first portion of a result set for the query while running the query according to the first join order, including adding to the result set a first record that matches the plurality of join operations in the query;

concurrent with running the query, collecting performance statistics about each of the join operations;

dynamically changing the first join order, during running of the query, to a second join order based on the statistics; ~~and~~

<u>where dynamically changing the first join order, during running of the query, to a second join order based on the statistics comprises:</u>

<u>determining respective fan-in statistics for the first join and second join; and</u>

<u>dynamically changing the first join order to the second join order if the respective fan-in statistics indicate that the second join is more likely to cause fan-in than the first join; and</u>

generating a second portion of the result set for the query while running the query according to the second join order, including adding to the result set a second record that matches the plurality of join operations in the query such that the result set includes at least the first and second records, the first and second portions of the result set generated for the same execution of the query.

2. (Canceled)

3. (Previously Presented) The method according to claim 1, further comprising the step of:

collecting additional statistics about each of the join operations after the first join order is changed to the second join order.

4. (Original) The method according to claim 3, further comprising the step of:

changing the second join order to either the first join order or a third join order based on the additional statistics.

5.-7. (Canceled).

8. (Currently Amended) The method according to claim ~~1~~ 5, further comprising the steps of:
~~determining respective fan-in statistics for the first join and second join;~~
determining respective fan-out statistics for the first join and the second join; and
dynamically changing the first join order to the second join order based on a combination of the respective fan-in and fan-out statistics.

9. (Previously Presented) The method according to claim 1, comprising the steps of:
identifying a predetermined sample size;
performing the step of collecting statistics for the predetermined sample size;
evaluating the collected statistics; and
dynamically changing the first join order to the second join order based on the collected statistics.

10. (Original) The method according to claim 9, comprising the steps of:
collecting additional statistics for substantially all of the query;
comparing the additional statistics with the collected statistics; and
adjusting the predetermined sample size, for use by a subsequent query, according to results of the comparing step.

11. (Previously Presented) The method according to claim 1, further comprising the steps of:
running another query after the query; and
selecting an initial join order for the other query based on the collected performance statistics.

12. - 23. (Canceled).

24. (New) A method for monitoring a query during runtime, said query involving a plurality of join operations, the method comprising the steps of:
wherein the plurality of join operations include a first join that includes a first table and a second table and a second join that includes the first table and a third table;
running the query according to a first join order;
generating a first portion of a result set for the query while running the query according to the first join order, including adding to the result set a first record that matches the plurality of join operations in the query;

concurrent with running the query, collecting performance statistics about each of the join operations;

dynamically changing the first join order, during running of the query, to a second join order based on the statistics;

where dynamically changing the first join order, during running of the query, to a second join order based on the statistics comprises:

determining respective fan-out statistics for the first join and the second join; and

dynamically changing the first join order to the second join order if the respective fan-out statistics indicate that the second join is less likely to cause fan-out than the first join; and

generating a second portion of the result set for the query while running the query according to the second join order, including adding to the result set a second record that matches the plurality of join operations in the query such that the result set includes at least the first and second records, the first and second portions of the result set generated for the same execution of the query.

Claims 5-7 canceled.
Claims 1, 8 amended.
Claim 24 added.

1. (Currently Amended) A method for monitoring a query during runtime, said query involving a plurality of join operations, <u>wherein the plurality of join operations include a first join that includes a first table and a second table and a second join that includes the first table and a third table,</u> the method comprising the steps of:

    running the query according to a first join order;

    generating a first portion of a result set for the query while running the query according to the first join order, including adding to the result set a first record that matches the plurality of join operations in the query;

    concurrent with running the query, collecting performance statistics about each of the join operations;

    dynamically changing the first join order, during running of the query, to a second join order based on the statistics; ~~and~~

    generating a second portion of the result set for the query while running the query according to the second join order, including adding to the result set a second record that matches the plurality of join operations in the query such that the result set includes at least the first and second records, the first and second portions of the result set generated for the same execution of the query<u>;</u>

    <u>determining respective fan-in statistics for the first join and second join; and</u>

    <u>dynamically changing the first join order to the second join order if the respective fan-in statistics indicate that the second join is more likely to cause fan-in than the first join.</u>

2. (Canceled)

3. (Previously Presented) The method according to claim 1, further comprising the step of:

    collecting additional statistics about each of the join operations after the first join order is changed to the second join order.

4. (Original) The method according to claim 3, further comprising the step of:

    changing the second join order to either the first join order or a third join order based on the additional statistics.

5.-7. (Canceled).

8. (Currently Amended) The method according to claim <u>1</u> ~~5~~, further comprising the steps of:

~~determining respective fan-in statistics for the first join and second join;~~

determining respective fan-out statistics for the first join and the second join; and

dynamically changing the first join order to the second join order based on a combination of the respective fan-in and fan-out statistics.

9. (Previously Presented) The method according to claim 1, comprising the steps of:

identifying a predetermined sample size;

performing the step of collecting statistics for the predetermined sample size;

evaluating the collected statistics; and

dynamically changing the first join order to the second join order based on the collected statistics.

10. (Original) The method according to claim 9, comprising the steps of:

collecting additional statistics for substantially all of the query;

comparing the additional statistics with the collected statistics; and

adjusting the predetermined sample size, for use by a subsequent query, according to results of the comparing step.

11. (Previously Presented) The method according to claim 1, further comprising the steps of:

running another query after the query; and

selecting an initial join order for the other query based on the collected performance statistics.

12. - 23. (Canceled).

24. (New) A method for monitoring a query during runtime, said query involving a plurality of join operations, wherein the plurality of join operations include a first join that includes a first table and a second table and a second join that includes the first table and a third table, the method comprising the steps of:

running the query according to a first join order;

generating a first portion of a result set for the query while running the query according to the first join order, including adding to the result set a first record that matches the plurality of join operations in the query;

concurrent with running the query, collecting performance statistics about each of the join operations;

dynamically changing the first join order, during running of the query, to a second join order based on the statistics; and

generating a second portion of the result set for the query while running the query according to the second join order, including adding to the result set a

second record that matches the plurality of join operations in the query such that the result set includes at least the first and second records, the first and second portions of the result set generated for the same execution of the query;

determining respective fan-out statistics for the first join and the second join; and

dynamically changing the first join order to the second join order if the respective fan-out statistics indicate that the second join is less likely to cause fan-out than the first join.